NETWORK PHYSICAL LAYER WITH EMBEDDED MULTI-STANDARD CRC GENERATOR

REFERENCE TO COMPACT DISC APPENDIX

[0001] The Compact Disc Appendix (CD Appendix), which is a part of the present disclosure, contains a hardware description language (Verilog code) description of receive and transmit modules of a network physical layer in accordance with an embodiment of the invention. A portion of the disclosure of this patent document contains material subject to copyright protection. The copyright owner of that material has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights.

[0002] The Ethernet local area network (LAN) is one of the most popular and widely used computer networks in the world. Since the early 1970's, computer networking companies and engineering professionals have continually worked to improve Ethernet product versatility, reliability, and transmission speeds. To ensure that new Ethernet products were compatible, interoperable, and reliable, the Institute of Electrical and Electronic Engineers (IEEE) formed a standards group to define and promote industry LAN standards. Today, the IEEE 802.3 standards group is responsible for standardizing the development of new Ethernet protocols and products under an internationally well-known LAN standard called the "IEEE 802.3 standard."

[0003] Currently, there are a wide variety of standard Ethernet products used for receiving, processing, and transmitting data over Ethernet networks. By way of example, these networking products are typically integrated into networked computers, network interface cards (NICs), SNMP/RMON probes, routers, switching hubs, bridges and

repeaters. To meet the demand for ever faster data transmission speeds, the IEEE 802.3 standards committee periodically introduces improved variations of the original IEEE 802.3 standard. For example, the "IEEE 802.3u standard" defines a system capable of transmitting data at speeds of up to about 100 Mbps and the "IEEE 802.3z standard" defines a system capable of transmitting data at speeds of up to 1 Gbps.

[0004] Figure 1 is a diagrammatic representation of a conventional open systems interconnection (OSI) layered model 100 developed by the International Organization for Standards (ISO) for describing the exchange of information between network layers. Though not all network standards follow OSI model 100-- Fibre Channel is a notable exception-- the OSI model is illustrative and useful for separating the technological functions of each layer.

[0005] OSI model 100 has as its lower-most layer a physical layer 105 that is responsible for encoding and decoding data into signals that are transmitted across a particular medium. As is well known in the art, physical layer 105 is also known as the "PHY layer." Above physical layer 105, a data link layer 110 provides reliable transmission of data over a network while performing appropriate interfacing with physical layer 105 and a network layer 115. Data link layer 110 generally includes a logical link control (LLC) layer 110A and a media access control (MAC) layer 110B. LLC layer 110A is generally a software function responsible for attaching control information to the data being transmitted from network layer 115 to MAC layer 110B. MAC layer 110B detects errors and schedules and controls the access of data to physical layer 105. In some cases, MAC layer 110B employs the well-known carrier sense multiple access with collision detection (CSMA/CD) algorithm. At the gigabit level and above, the CSMA/CD function has essentially been eliminated. MAC layer 110B is optionally connected to physical layer 105 via a

Gigabit Medium Independent Interface (GMII).

[0006] Like data link layer 110, physical layer 105 includes multiple sublayers. A physical coding sublayer (PCS) 105A synchronizes and reformats data frames from link layer 110 into 10-bit code groups. A physical medium attachment (PMA) sublayer 105B serializes and transmits the code groups. PMA sublayer 105B deserializes data coming in from a communication medium 140 via a medium-dependent interface (MDI) and a physical medium dependent (PMD) layer 105C, and is additionally responsible for recovering the clock from incoming data streams.

[0007] Network layer 115 routes data between nodes in a network, and initiates, maintains, and terminates a communication link between users connected to those nodes. Transport layer 120 performs data transfers within a particular level of service quality. By way of example, a typical software protocol used for performing transport layer 120 functions may be TCP/IP, Novell IPX and NetBeui. Session layer 125 controls when users are able to transmit and receive data depending on whether the user is capable of full-duplex or half-duplex transmission, and also co-ordinates between user applications needing access to the network. Presentation layer 130 is responsible for translating, converting, compressing and decompressing data being transmitted across a medium. As an example, presentation layer 130 functions are typically performed by computer operating systems like Unix, DOS, Microsoft Windows, and Macintosh OS. Finally, Application layer 135 provides users with suitable interfaces for accessing and connecting to a network.

[0008] For more information on Ethernet network communication technology, reference may be made to issued U.S. Patents entitled "Apparatus and Method for Full-Duplex Ethernet Communications" having U.S. Pat. Nos. 5,311,114 and 5,504,738, and "Media Access Control Micro-RISC Stream Processor and Method for Implementing the Same" having U.S.

Pat. No. 6,172,990. These patents are incorporated herein by reference.

[0009] Figure 2 is a flowchart 200 depicting the operation of portions of link layer 110 and physical layer 105 of Figure 1 when transmitting a data frame. Beginning at step 205, link layer 110 assembles data received from network layer 115 to create a data frame 210. Frame 210 generally includes a seven-byte preamble followed by a single-byte start frame delimiter (SFD). After the start frame delimiter, a six-byte destination address DA identifies the node that is to receive frame 210. A source address SA -- also six bytes -- follows the destination address DA. Next, a type/length field (typically 2 bytes) indicates the length and type of a data/pad field that follows. As is well known in the art, if a length is provided, the frame is classified as an 802.3 frame, and if the type field is provided, the frame is classified as an Ethernet frame. The data/pad field contains the data from network layer 115 divided into a sequence of octets (The word "octet" is an Ethernet word, also referred to as a "byte"). Correct CSMA/CD protocol requires a minimum frame size, which is specified by the particular implementation of the standard. If necessary, the data field is extended by appending extra bits (that is, a "pad") in units of octets after the data field.

[0010] Moving to step 215, link layer 110 performs a thirty-two-bit cyclic redundancy check (CRC) to calculate a CRC value. The CRC value is a function of the contents of frame 210 except for the preamble, SFD, FCS, and extension fields. The CRC value is then appended to frame 210 in a frame check sequence (FCS) field. Next, before passing the frame on to physical layer 105, the link layer optionally adds an extension field, which enforces the minimum carrier event duration in some operational modes.

[0011] PCS sublayer 105A accepts frame 210 from link layer 110 and encapsulates frame 210 (step 225) into a

packet 227. In the art, packets like packet 227 are often referred to as "physical layer streams." In the present disclosure, the term "physical layer stream" refers to sequences of packets 227.

[0012] Properly formed, packet 227 includes a Start-of-Stream Delimiter (SSD), data code groups (DATA) corresponding to the data from the link layer, and an End-of-Stream Delimiter (ESD) (In some standards, the ESD can be replaced by a special SSD that can perform multiple functions. In the present disclosure, the placement of the delimiter defines whether it is a start-of-frame or end-of-frame delimiter). In addition, some standards specify that idle data IDLE be included in a physical layer stream between some packets 227. Each packet and associated idle data are collectively termed a "packet assembly" 234 for purposes of this disclosure.

[0013] The PCS sublayer calculates the running disparity for each packet assembly (step 235). Running disparity maintains an equivalence between the number of transmitted ones and zeros to keep the DC level balanced halfway between the "one" voltage level and the "zero" voltage level. Running disparity can be either positive or negative. In the absence of errors, the running disparity value is positive if, since power-on or reset, more ones have been transmitted than zeros, and is negative if more zeros have been transmitted than ones. The PCS sublayer adjusts the disparity and provides the disparity-adjusted physical layer stream to the PMA sublayer.

[0014] The entire link layer 110, and sometimes portions of physical layer 105, can be implemented using configurable logic in a programmable logic device (PLD), commonly a field-programmable gate array (FPGA). (For a more detailed treatment of one such embodiment, see the Xilinx Product specification entitled "1-Gigabit Ethernet MAC Core," November 28, 2001, which is incorporated herein by reference.) Unfortunately, though a relatively simple

function, the CRC circuitry in the link layer can occupy a significant portion of the available programmable resources, leaving fewer resources for other tasks. There is therefore a need for a more efficient means of facilitating network functionality in programmable logic.

SUMMARY

[0015] The present invention is directed to methods and structures for transmitting and receiving data over a network. In an embodiment consistent with the OSI network model, the transmit and receive CRC generators are moved from the link layer to the physical layer. This modification frees up valuable programmable logic resources when a programmable logic device is employed to perform the functions of the link layer.

[0016] In one embodiment, the CRC generators of the physical layer are adapted to comply with a plurality of network communication standards. In yet another embodiment, the physical layer, including the CRC generators, is instantiated in hard logic on a programmable logic device. This embodiment offers a flexible and efficient solution for providing the physical and link layers on a single integrated circuit.

[0017] This summary does not define the scope of the invention, which is instead defined by the appended claims.

BRIEF DESCRIPTION OF THE FIGURES

[0018] Figure 1 is a diagrammatic representation of a conventional open systems interconnection (OSI) layered model 100 for describing the exchange of information between network layers.

[0019] Figure 2 is a flowchart 200 depicting the operation of portions of link layer 110 and physical layer 105 of Figure 1 when transmitting a data frame.

[0020] Figure 3 depicts a portion of a network transmitter 300 in accordance with one embodiment of the

invention.

[0021] Figure 4 is a flow chart 400 describing the sequence of steps performed by the link layer and the physical layer of an embodiment that complies with the Gigabyte Ethernet standard.

[0022] Figure 5 depicts a portion of a network receiver 500 in accordance with one embodiment of the invention.

[0023] Figure 6 depicts an FPGA 600 adapted in accordance with an embodiment of the invention to include network transmitter 300 of Figure 3 and network receiver 500 of Figure 5.

DETAILED DESCRIPTION

[0024] Figure 3 depicts a portion of a network transmitter 300 in accordance with one embodiment of the invention. Transmitter 300 only depicts a data link layer 305 and PCS layer 310 modified in accordance with embodiments of the present invention; the remaining layers and sublayers are identical to those discussed above in connection with Figures 1 and 2.

[0025] Data link layer 305 is, like the prior art, adapted to receive data from network layer 115 via an LLC sublayer 110A (Figure 1). Data link layer 305 additionally includes a MAC sublayer 315 that does not calculate a CRC as is done in conventional MAC sublayers; instead, as will be discussed below in detail, CRC functions required by different network standards are performed in the physical layer by a modified PCS 310. In the depicted embodiment, link layer 305 is instantiated in programmable logic 316, but all or a portion may be "hardwired."

[0026] PCS 310 includes a data encapsulator 317 that encapsulates frames from MAC sublayer 315 in the manner described above in connection with Figure 2. The encapsulation performed by data encapsulator 317 reformats frames into packets and, for some packets, inserts idle data. As noted previously, packets with associated idle

data are collectively referred to herein as a "packet assembly." In the depicted embodiment, data encapsulator 317 is instantiated in programmable logic 316 with link layer 305, but data encapsulator 317 might also be hardwired.

[0027] In an embodiment that complies with the IEEE 802.3z standard, the idle data is a two-byte sequence in which the first byte is a K28.5 "comma" defined by the standard and the second byte renders the sequence either correcting or non-correcting. However, because the idle data depends upon the non-existent CRC value, in one embodiment data encapsulator 350 merely inserts, by default, the correcting form (or non-correcting form) of the idle data.

[0028] PCS 310, with the exception of data encapsulator 317, is instantiated in hard logic 319. PCS 310 includes a CRC generator 318, which in turn includes a programmable demultiplexer 320 adapted to provide the output of data encapsulator 317 to any of a number of data ports within PCS 310. Demultiplexer 320 can be programmed using memory cells (not shown) such as those commonly available on programmable logic devices.

[0029] CRC generator 318 additionally includes a CRC module 325 that receives data frames modified to comply with a number of communication standards. In the depicted embodiment, packets and packet assemblies from data encapsulator 317 can be routed via demultiplexer 320 to four different modules, each of which modifies the function of CRC module 325 to comply with a particular standard. The four modules include an InfiniBand™ module 330, a Gigabit Ethernet module 335, a Fibre Channel module 340, and a User-Mode module 345. Depending on the selected communication standard, as determined by programming demultiplexer 320, CRC module 325 calculates a CRC for each incoming frame embedded in a packet from data encapsulator 317 and inserts the resulting CRC value into the appropriate FCS field of

the packet derived from the frame. InfiniBand™ module 330 works with CRC module 325 to perform a CRC in compliance with the specification entitled "InfiniBand™ Architecture Release 1.0.a," June 19, 2001; Gigabit Ethernet module 335 works with CRC module 325 to perform a CRC in compliance with the IEEE 802.3z Gigabit Ethernet specification; and Fibre Channel module 340 works with CRC module 325 to perform a CRC in compliance with the Fibre Channel standard, as outlined in "Fibre Channel Overview," by Zoltán Meggyesi of the Research Institute for Particle and Nuclear Physics. Each of the foregoing documents is incorporated herein by reference. User module 345 can be adapted to perform a CRC in compliance with e.g. another standard.

[0030] CRC module 325 includes an optional force-error input line FE connected to an external, user-accessible pin (not shown). If line FE is held to a logic zero, module 325 provides the CRC value to data pipe 355 as described above. If, on the other hand, line FE is held to a logic one, module 325 corrupts the last byte of the CRC value to force a CRC error. Force-error line FE and related circuitry allow users to verify the operation of CRC module 325.

[0031] In one embodiment, module 325 corrupts the CRC value by XORing each bit of the last byte of the CRC value with a logic one to produce a corrupt CRC value in which each bit of the last byte is inverted. In another embodiment, users can configure inputs to the XOR function to be either ones or zeros, and can therefore determine which bits of the last byte are inverted. In still other embodiments, the last byte is replaced with a fixed value or one of two or more alternative values.

[0032] Positioning CRC generator 318 within the physical layer in hard logic minimizes the amount of circuitry required to cover multiple standards. CRC module 325 is reasonably similar for each of the standards, only requiring minor modifications via modules 330, 335, 340, and 345. For example, the Gigabit Ethernet standard runs all frame bits

through CRC module 325 to create a CRC value, while the InfiniBand™ standard, depending upon the packet, masks out some bits before performing the CRC. The different modules account for such differences, but each relies on the same function performed by CRC module 325.

[0033] PCS 310 cannot determine whether to send the correcting or the non-correcting form of the idle data until the disparity for the entire packet assembly is known, but the disparity cannot be calculated until the CRC value is in place within the packet. Data pipe 355 receives the packet assembly, sans the CRC value, from data encapsulator 317 and then inserts into the FCS field of the associated packet the CRC value calculated by CRC module 325. Data pipe 355 then conveys the packet assembly with the potentially erroneous idle data to packet-assembly modifier 360. 8B/10B encoder 365 calculates the running disparity on the resulting packet assembly and conveys the disparity to assembly modifier 360, which modifies the packet assembly, if necessary, to provide the appropriate one of the correcting or non-correcting forms. In the case of a system employing the Gigabyte Ethernet standard, the disparity should be negative before transmitting data from the physical layer, so packet-assembly modifier 360 modifies the idle data to the correcting form in the event that the disparity is positive.

[0034] Figure 4 is a flow chart 400 describing the sequence of steps performed by link layer 305 and PCS sublayer 310 of an embodiment that complies with the Gigabyte Ethernet standard. Link layer 305 assembles each frame received from the MAC client (step 205) in the manner described above in connection with Figure 2. Link layer 305 also adds an extension field (step 220), also in the manner discussed above. Different from the process described above, link layer 305 does not calculate a CRC value for insertion in the FCS field of frame 405. Instead, MAC sublayer 305 sends the frame without a CRC value, and with or without a CRC field. In one embodiment, MAC sublayer 305

adds four extra bytes onto the frame and then sends the frame normally. In this case, the four extra bytes are merely placeholders for the CRC: their contents do not matter.

[0035] Next, data encapsulator 317 encapsulates the resulting frame 405 in the manner discussed above in connection with Figure 2 (step 225) and appends idle data to the resulting packet 410 to form a packet assembly 411. The correct form of the idle data must be "guessed," because the Gigabyte Ethernet standard requires the idle data be a function of the CRC value, and the CRC value has yet to be calculated. CRC module 325 inserts the calculated CRC value into packet 410 (step 415). In the Gigabit Ethernet standard, the idle data comprises a two-byte sequence in which the first byte is a K28.5 "comma" character and the second byte makes the sequence correcting or non-correcting. The K28.5 comma can be positive (bit sequence 0011111010) or negative (bit sequence 1100000101). Encoder 365 sends the positive comma if the disparity is negative, and sends the negative comma if the disparity is positive. However, minus commas or sequences of minus commas are not recognized by many Gigabit-Ethernet compliant devices, and thus should be sent as seldom as possible. For example, the Gigabit Ethernet specification requires the minus comma be sent at most once per collection of idle data. Subsequent commas associated with the same packet assembly must be plus commas. Consequently, packet-assembly modifier 360 determines whether the disparity is positive (decision 420) and, if so, modifies packet assembly 411 to include a correcting form of the idle data (step 425). If the disparity is not positive, then packet-assembly modifier 360 leaves the idle data as is. In either case, encoder 365 conveys the resulting correct packet assembly to the PMA sublayer (step 435). The remaining transmission sequence is conventional, and is therefore omitted for brevity.

[0036] For more detailed discussion of link and physical

layers of the prior art, see IEEE standard 802.3, 2000 edition, entitled, "Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," which is incorporated herein by reference.

[0037] The Fibre Channel standard also discourages transmission of "negative commas." Instead of modifying the idle data, however, modifier 360 is adapted to modify the ESD field of packet 410 to correct for a positive disparity. The InfiniBand™ standard does not require either an idle modifier or an end-of-packet modifier.

[0038] Figure 5 depicts a portion of a network receiver 500 in accordance with one embodiment of the invention. Receiver 500 only depicts a data link layer 505 and a PCS layer 510 modified in accordance with embodiments of the present invention; the remaining layers and sublayers are identical to those discussed above in connection with Figures 1 and 2. In one embodiment, data link layers 305 and 505 are portions of the same link layer, and PCS sublayers 310 and 510 are portions of the same PCS sublayer.

[0039] PCS sublayer 510 includes a 8B/10B decoder 515, an elastic buffer 520, a CRC generator 525, and a data decapsulator 530. In the depicted embodiment, all these elements except for data decapsulator 530 are instantiated in hard logic 533, though this need not be the case.

[0040] 8B/10B decoder 515 (sometimes referred to as an "10B/8B decoder") conventionally receives and decodes data from a PMA sublayer and conveys the resulting decoded packet assemblies to elastic buffer 520. Also conventional, decoder 515 identifies some types of packet errors and alerts MAC sublayer 570 of erroneous packets via e.g. an error line 534.

[0041] Elastic buffer 520 is a conventional buffer with adjustable data capacity; in one embodiment, buffer 520 can hold up to 64 bytes of data, an amount sufficient to comply with each of the above-mentioned standards. Buffer 520

forwards packet assemblies to CRC generator 525 and data decapsulator 530.

[0042] CRC generator 525 includes a programmable demultiplexer 535 that provides packet assemblies from buffer 520 to any of a number of data ports within PCS 510. PCS 510 additionally includes a CRC module 540 that receives data from one of four sources. In the depicted embodiment, packet assemblies from buffer 520 can be routed via demultiplexer 535 to four different modules, each of which modifies the function of CRC module 540 to comply with a particular standard. The four modules include an InfiniBand™ module 545, a Gigabit Ethernet module 550, a Fibre Channel module 555, and a User-Mode module 560. Depending on the selected communication standard, as determined by programming demultiplexer 535, CRC module 540 calculates a CRC value for each incoming packet assembly. This CRC value depends on the same fields for which the previously mentioned CRC value was calculated in the foregoing discussion of Figures 3 and 4.

[0043] CRC generator 525 includes a CRC compare module 565 that strips the CRC value from each packet assembly and compares the stripped CRC value with the calculated CRC value from CRC module 540. During the comparison process, CRC compare module 565 alerts link layer 505 by asserting a signal "checking CRC." In the event of a mismatch between the stripped and calculated CRC values, CRC compare module 565 generates an error signal to link layer 505 by pulling a line CRC INVALID high (i.e., to a logic one).

[0044] Data decapsulator 530 conventionally strips headers and removes idle data from incoming packet assemblies to reproduce data frames. The frames are then conveyed to a MAC sublayer 570 within link layer 505. As with MAC sublayer 315 of transmitter 300 (Figure 3), MAC sublayer 570 does not calculate a CRC value; instead, as noted above, the CRC functions required by different network standards are performed in hard logic in PCS sublayer 510.

Similar to the transmitter case, positioning CRC module 525 within the physical layer minimizes the amount of programmable resources required to implement the CRC function.

[0045] As is conventional, MAC sublayer 570 "flushes" erroneous packets, whether those packets are identified by decoder 515 or by a CRC mismatch. Unlike conventional MAC sublayers, however, MAC sublayer 570 relies upon CRC generator 525 to find CRC errors. In the depicted embodiment, MAC sublayer 570 has no control over whether CRC module 525 performs a CRC on incoming packets, so receiver 500 performs a CRC on each packet regardless of whether decoder 515 identifies an error. The absence of MAC-sublayer control places CRC generator 525 outside of the conventional boundary of the link layer.

[0046] Figure 6 depicts an FPGA 600 adapted in accordance with an embodiment of the invention to include network transmitter 300 of Figure 3 and network receiver 500 of Figure 5. As is conventional, FPGA 600 includes a collection of programmable logic, including a plurality of input/output blocks (IOBs) 605, an array of configurable logic blocks (CLBs) 610, and a plurality of block RAMs 615. CLBs 610 are the primary building blocks and contain elements for implementing customizable gates, flip-flops, and wiring; IOBs 605 provide circuitry for communicating signals with external devices; and block RAMs 615 allow for synchronous or asynchronous data storage, though each CLB can also implement synchronous or asynchronous RAMs. Some of IOBs 605 may be optimized, as necessary, to support high-speed communication. For a detailed treatment of one FPGA, see the Xilinx advance product specification entitled "Virtex-II 1.5V Field-Programmable Gate Arrays," DS031-2 (v1.9), November 29, 2001, which is incorporated herein by reference.

[0047] In addition to conventional features, FPGA 600 includes hardwired (i.e., application specific) logic 319

(Figure 3) and 533 (Figure 5), which respectively include CRC generators 318 and 525. Data link layer 305 and data link layer 505 are instantiated within programmable logic 316 and 575, respectively, using a plurality of CLBs 610.

[0048] While the present invention has been described in connection with specific embodiments, variations of these embodiments will be obvious to those of ordinary skill in the art. For example, many of the elements instantiated in programmable logic can be instantiated instead in hard logic, and vice versa. Therefore, the spirit and scope of the appended claims should not be limited to the foregoing description.